Lab 5 – Feed-Forward Artificial Neural Networks

Evolution and Learning in Computational and Robotic Agents
MSE 2400 Dr. Tom Way

Introduction

In this lab you will build several feed-forward neural networks and test how well they work (that is, develop learning agent models) on various training input domains. You will also gain some experience in building your own training set for neural networks. Finally, you will gain some first-hand understanding of the limits of the training of a neural network and you will observe how the network "generalizes."

The software we will use for this lab is for Windows computers, and the work to be done is ample enough, that we will work in teams of 1, 2 or 3 students each. If you have a Mac and do not have a way to run Windows programs on your Mac, then make sure to team up with somebody who has a Windows laptop. You can work in a team of up to three people.

Note that significant exploration of the software, how it works, what it can do, and how it can be used likely will be required.

Worth

- 100 points

Due

- Your lab report will be due the **first regular lecture class meeting after your team completes the lab**.
- This lab contains a lot of steps, so it may take 2 lab sessions to complete. In addition, you may want to spend time outside of class to work on the lab exercises and writing of the lab report.

What to hand in

- A typewritten lab report, containing answers to all of the questions in **Part 4** and **Part 5** of this lab instruction document.
- Make sure the names of all team members are on the report.
- You can hand in the report via email or on paper.

Lab Steps

Part 1 - Software Setup

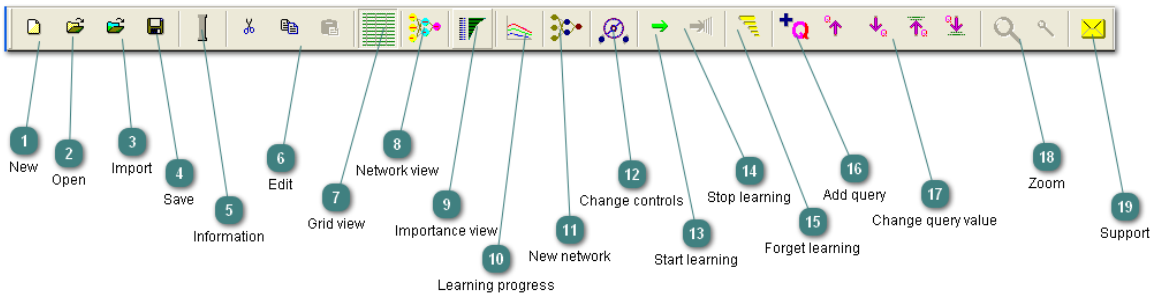
1. Go to JustNN.com website and download the JustNN software package.
2. Install it on the laptop your team will be using for the lab by clicking on the installer you downloaded. Accept all the default prompts, and let it install in the folders the installer asks to install in.
3. After installation, find the desktop icon for JustNN and double-click on it to start the program. If it starts successfully, you can exit it and move on to Part 2.

Part 2 - Software Orientation (JustNN User Manual Excerpts)

If you are comfortable with the JustNN software, you can skip this part, otherwise review this part to learn more about JustNN. This part will be easier to read an understand in the online version of this handout, as you can see the colors of the icons and menu options.

In the next part of this lab you will run through three sample exercises with JustNN to get some familiarity with the software.

When you start the JustNN system, you'll get a window with the following toolbar just below the standard "Windows Pulldown Menus" at the top of the window:



Here is a description of the toolbar buttons you will use:

1 **New**

Create a new document with a blank neural network grid

2 **Open**

Open an existing grid and neural network.

3

Import

Import a TXT, CSV, XLS, BMP or binary file into the neural network grid.

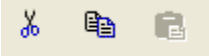
4

Save

Save the active neural network document. This will contain both the training data and the network you build from it.

6

Edit



Cut, copy and paste in the Grid. The row or column is selected by double clicking the row or column name.

7

Grid view

View and edit the Grid. The Grid is like an Excel spreadsheet. Each row is an example or query for the Network. (see Grid View, below)

8

Network view

View the neural Network that is created from the data in the Grid.

9

Importance view

View the importance of the inputs (e.g. a list of the network's inputs in descending order of weight).

10

Learning progress

View the learning progress graphs. The graphs show the minimum, maximum, and average error rates as the network is trained on more and more cycles (iterations over the **backpropagation** algorithm). If you provide validating examples (examples that are not used for training, just testing), the graph will also show average validating error vs. learning cycles.

11

New network

Opens the New network dialog to create a neural network from the Grid.

13 **Start learning**

Starts the learning process. Pressing this will make the network iterate over the Backpropagation algorithm (we discussed in class) and update the weights in the network to make it respond correctly for all the training examples in the Grid.

14 **Stop learning**

Stop the learning process.

15 **Forget learning**

Forget learning. The network's weights are all reset to random number values between -0.5 and +0.5.

16 **Add query**

Adds a querying row to the Grid.

17 **Change query value**

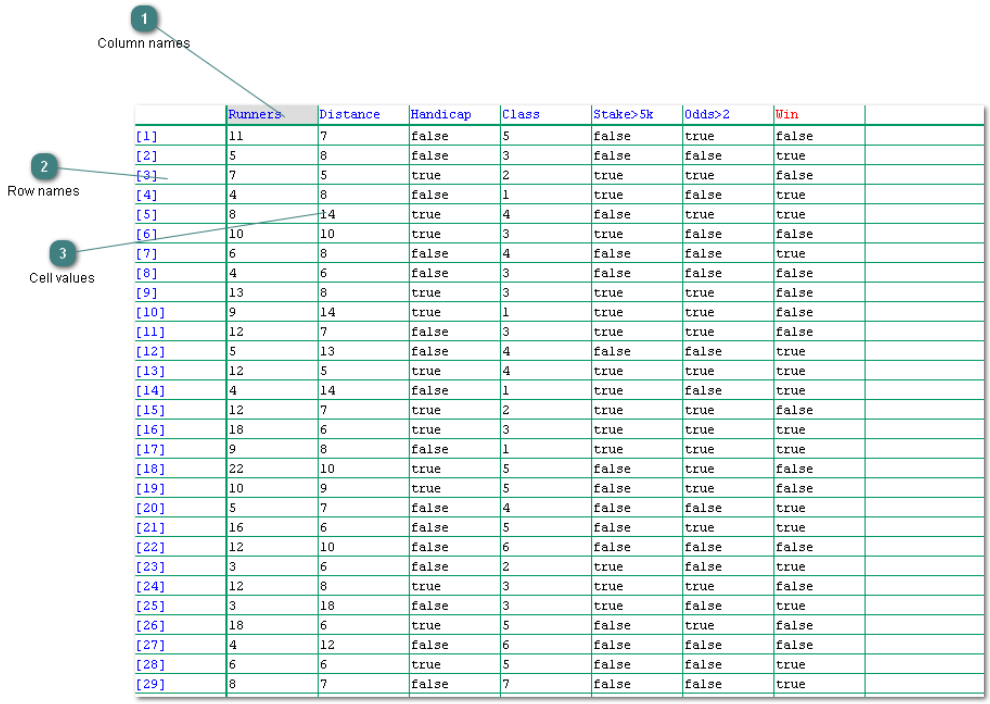
Increases, decreases, maximises or minimizes the query value.

2.A - Grid View

Now let's look at the two main views provided by JustNN that you'll use in this project. First, the Grid (see next page). You can view it by pressing the button in the toolbar.

The Grid view shows all the Examples arranged in rows and all the Input/Outputs arranged in columns. Input is the same as a feature. The first column contains the Example types and names. The first row contains the Input/Output types and names.

Everything on the Grid can be edited by moving to the cell containing the value and then pressing the enter key to start the Edit Grid dialog. The cell can be selected either using the arrow keys or the mouse. A single click will select the cell and a double click will start the Edit Grid dialog. A double click on the Example name cell will select the whole row and a double click on the Input/Output name cell will select the whole column. The row or the column can be deselected by pressing the Esc key.



	Runners	Distance	Handicap	Class	Stake>5k	Odds>2	Win
[1]	11	7	false	5	false	true	false
[2]	5	8	false	3	false	false	true
[3]	7	5	true	2	true	true	false
[4]	4	8	false	1	true	false	true
[5]	8	14	true	4	false	true	true
[6]	10	10	true	3	true	false	false
[7]	6	8	false	4	false	false	true
[8]	4	6	false	3	false	false	false
[9]	13	8	true	3	true	true	false
[10]	9	14	true	1	true	true	false
[11]	12	7	false	3	true	true	false
[12]	5	13	false	4	false	false	true
[13]	12	5	true	4	true	true	true
[14]	4	14	false	1	true	false	true
[15]	12	7	true	2	true	true	false
[16]	18	6	true	3	true	true	true
[17]	9	8	false	1	true	true	true
[18]	22	10	true	5	false	true	false
[19]	10	9	true	5	false	true	false
[20]	5	7	false	4	false	false	true
[21]	16	6	false	5	false	true	true
[22]	12	10	false	6	false	false	false
[23]	3	6	false	2	true	false	true
[24]	12	8	true	3	true	true	false
[25]	3	18	false	3	true	false	true
[26]	18	6	true	5	false	true	false
[27]	4	12	false	6	false	false	false
[28]	6	6	true	5	false	false	true
[29]	8	7	false	7	false	false	true

In the picture above, the first row represents a single training example to be fed to the network. It has the first Input/Feature “Runners”, which has the value 11, the second Input/Feature “Distance,” which has the value 7, etc. Note the last value in the row, “Win,” is color coded in red. It is an Output. That is, this is what the network is supposed to return when the example is fed into it. In other words, it is the Ideal Output for this example.

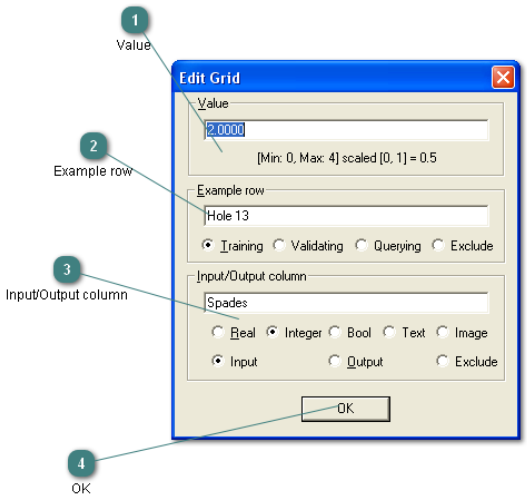
2.A.1 Creating a New Grid.

A new Grid is created by pressing the toolbar button or using the **File > New** menu command. The new Grid will be empty except for a horizontal line, a vertical line and an underline marker that shows the current position in the Grid. New Grid rows and columns are created at the current position.

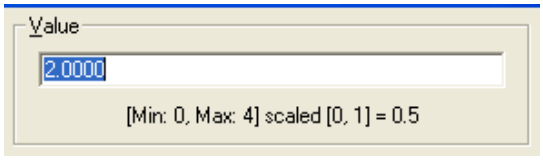
2.A.2 Creating the first Example row and Input/Output column.

Press return and a prompt will appear that says "Create new Example row?". Answer Yes. Another prompt will appear that says "Create new Input/Output column?". Answer Yes again. You will now see that the Grid has one cell containing "?", a row name containing "T:0" and a column name containing "I:0". The "?" indicates that the cell has no value, the "T:0" indicates that it is a Training Example in row 0 and the "I:0" indicates that is is an Input in column 0. Press return again and an Edit Grid dialog box will appear that allows you to enter the cell value. Using the same dialog you can change the Input/Output column name, mode and type. The dialog can also be used to change the Example row name and type.

Here is an example of the Edit Grid Dialog Box, and an explanation of what its fields mean. In the lab projects, you will be using Input/Output column types (bottom of box) that are either Bool (Boolean) or Image. This box appears every time you edit a cell in the Grid.

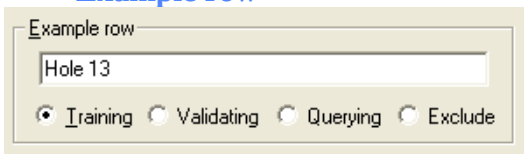


1 Value



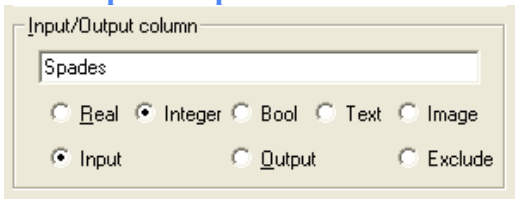
The value in the selected cell with minimum, maximum and scaled value.

2 Example row



Enter the example row name and set the type of row.

3 Input/Output column



Enter the column name. Set the column type and mode (Input or Output).

OK

Click to complete the edit and close the dialog.

2.A.3 Creating more Input/Output columns.

Move the marker one cell to the right by pressing the right arrow or tab key. Now press the return key and a prompt will appear that says "Create new Input/Output column?". Answer Yes. Press return again and the Edit Grid dialog box will appear. This time the Example row will already contain a name and type.

You can set the cell value, the Input/Output name and the type can be set to "I:" for input, "O:" for output, "X:" for exclude or "S:" for serial.

The mode can be set to "Real", "Integer", "Bool" or "Text".

Any type of Input/Output column can be inserted into the Grid using the functions on the Insert menu.

2.A.4 Creating more Example rows.

Move the marker one cell down by pressing the down arrow key. Now press the return key and a prompt will appear that says "Create new Example row?". Answer Yes. Press return again and the Edit Grid dialog box will appear. This time the Input/Output column will already contain a name and type.

You can set the cell value, Example name and the type can be set to T: for training, V: for validating, Q: for querying or X: for exclude. Any type of Example row can be inserted into the Grid using the functions on the Insert menu.

2.A.5 Copying Example rows and Input/Output columns.

Double click on the name to select the whole row or column. Cut will remove the selected row or column and place it on the clipboard. Copy will place a copy of the selected row or column on the clipboard. Paste will insert a copy of the clipboard before the currently selected row or column.

If the clipboard contains a row then Paste will insert the row into the Grid. If the clipboard contains a column then Paste will insert a column into the Grid. The invisible Grid data, limits and defaulted values, will be regenerated after a Paste column thus any neural network that has already been generated from the Grid will be invalidated.

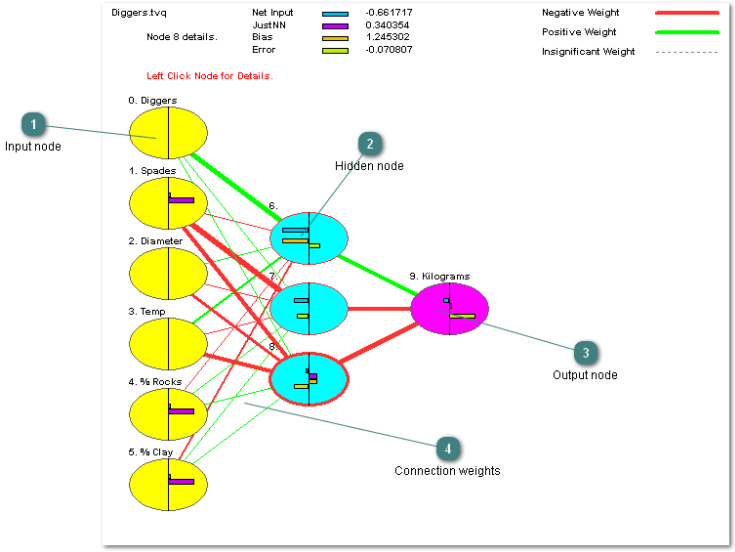
2.A.6 Query Example Rows

As stated in 3.A.4, you can make a Grid row represent a Query when you create it. That is, it will be an example that is fed into a **trained** network. You won't have to edit the output cell(s) of this example. They will be automatically set by the action of the trained neural network.

Be sure you don't get confused between Query example rows and Training Example rows! Similarly, Validating example rows represent examples that you do not want to train the network on, only test the network once it is trained on all the training examples.

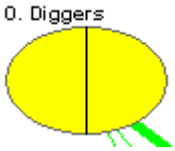
2.B - Network View

Now let's take a look at the view of the network that is created from the data in the Grid. Open the Sample "Diggers.tvg". You view it by pressing the button in the toolbar.



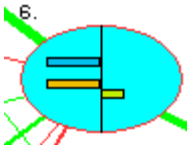
The **Network** view shows how the nodes in a **JustNN** neural network are interconnected.

1 **Input node**



This represents the input/feature value being read in from the Grid. There are always as many input nodes as there are features/inputs for an example in the Grid.

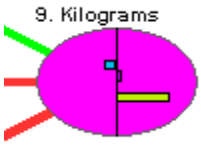
2 **Hidden node**



Hidden nodes are fully connected to input nodes, output nodes or other layers of hidden nodes.

3

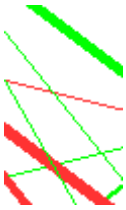
Output node



Output nodes are connected to the output columns in the grid.

4

Connection weights



The input layer is fully connected to the first hidden layer. Each connection has a weight that is updated while the network is learning. Hidden layers are fully connected to the next hidden layer or the output layer. Red connections represent weights with **negative** values, and green connections represent weights with **positive** values. The thinner or thicker the connection line, the smaller or larger the absolute value of the weight is. (Dashed lines represent weights extremely close to 0)

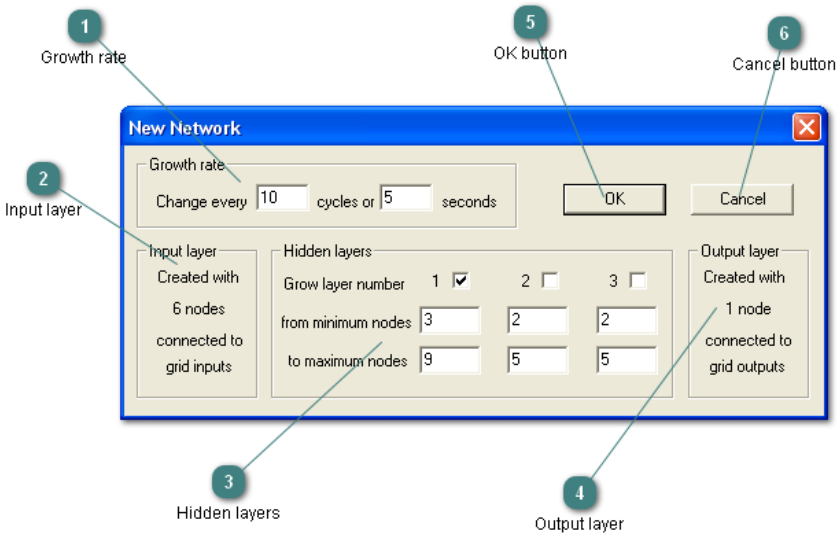
2.B.1 How to create a new neural network

A new neural network can be created from the Grid by pressing the **New Network** toolbar button or selecting **Action > New Network**.

This will produce the New Network dialog. This dialog allows the neural network configuration to be specified. The dialog will already contain the necessary information to generate a neural network that will be capable of learning the information in the Grid.

However, the generated network may take a long time to learn and it may give poor results when tested. A better neural network can be generated by checking “Grow Hidden Layer 1” in the Create Netwrok Dialog box and allowing JustNN to determine the optimum number of nodes and connections.

When you create a new network, you’ll see a “New Network” dialog box, followed by a “Controls” dialog box to set up things like the network’s learning rate (alpha), validation process, and when to automatically stop training (so we don’t get caught in infinite learning when convergence never happens). Let’s take a look at the “New Network” box:



1

Growth rate

A network is produced when the cycles or seconds elapses until the optimum network is found.

2

Input layer

The number of nodes in the input layer is determined by the number of input columns in the grid.

3

Hidden layers

The hidden layers are grown from the minimum to the maximum number of nodes. Always make sure Grow Layer Number 1 is checked.

4

Output layer

The number of nodes in the output layer is determined by the number of output columns in the grid.

5

OK button

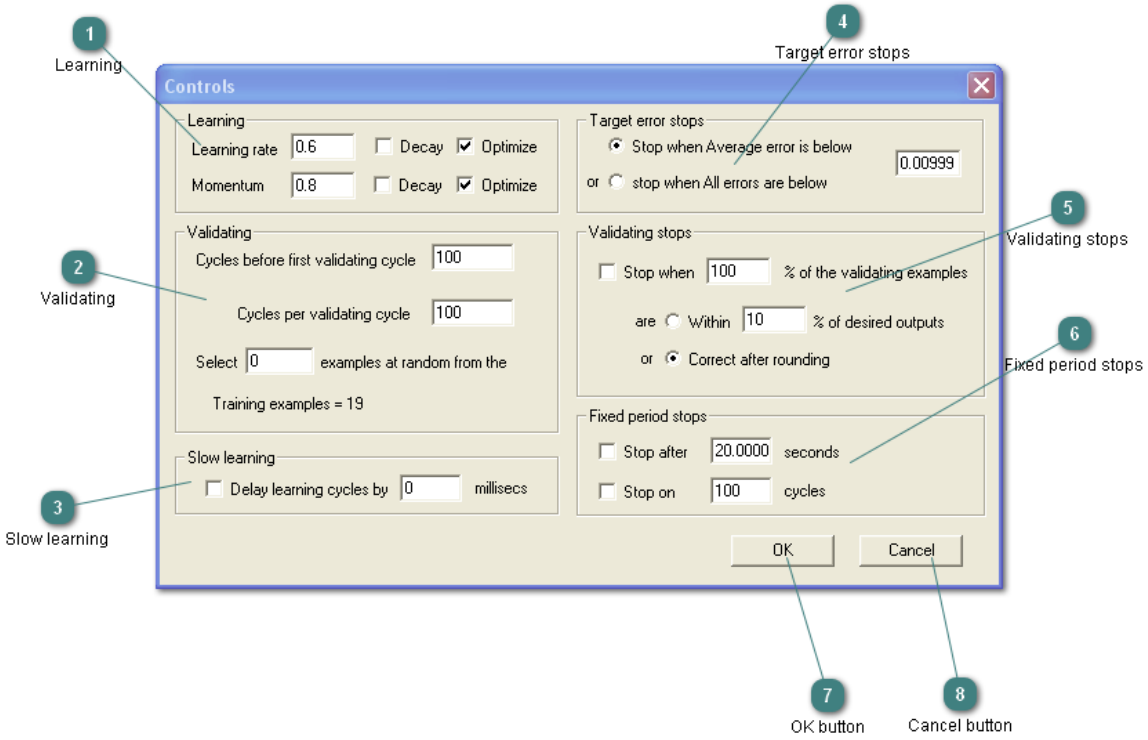
Press to accept all the settings and close the dialog.

6

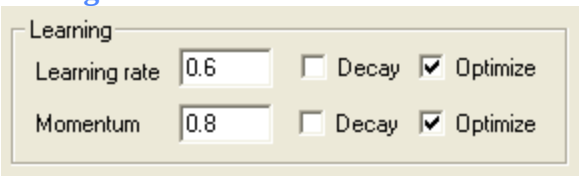
Cancel button

Press to reject all the settings and close the dialog.

Now let's take a look at the "Controls" Dialog Box:

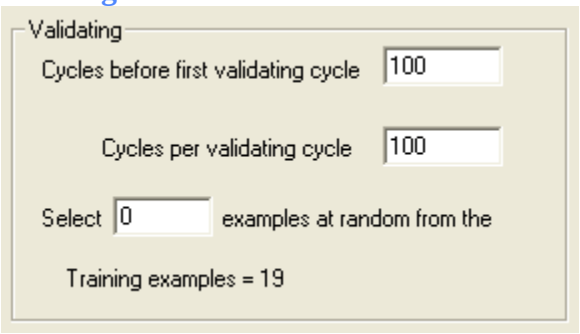


1 Learning



Always make sure for our labs that you check the Optimize box!

2 Validating


Validating
Cycles before first validating cycle 100
Cycles per validating cycle 100
Select 0 examples at random from the
Training examples = 19

Typically, if you have no Validating examples in the Grid, set this to select a number that is no more than 25% of the training examples.

3

Slow learning

Ignore this for our labs.

4

Target error stops

Accept what it already shows, for our labs.

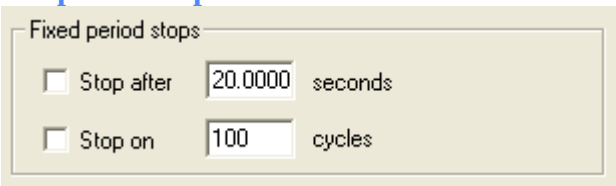
5

Validating stops

Accept what it says.

6

Fixed period stops



Select the "Stop On" option for our labs, and specify 200 or 1000 cycles. Reasonable values are 200 for the projects in Part 4 of the Lab, and 1000 for Part 5 (the images of bills).

7

OK button

Press to accept all the settings and close the dialog.

8

Cancel button

Press to reject all the settings and close the dialog.

Part 3 - JustNN Exercises

You will now run three exercise projects on JustNN, to familiarize yourself with the interface, learn how to create training example sets on the Grid, learn how to edit the Grid, learn how to create networks, and learn how to query them. These can be started by clicking the **Getting Started** button on the **Tip of the Day** or using the menu command **Help > Getting Started...**

The network files you'll use to work with these exercises are already in the JustNN folder the Installer created for you. If you find you have to save one of these network files after you make changes to them, please save them as a different file, otherwise you won't be able to go back to the exercise if you want to check things out again.

YOU DO NOT HAVE TO HAND IN ANYTHING FROM THESE EXERCISES FOR THE LAB REPORT.

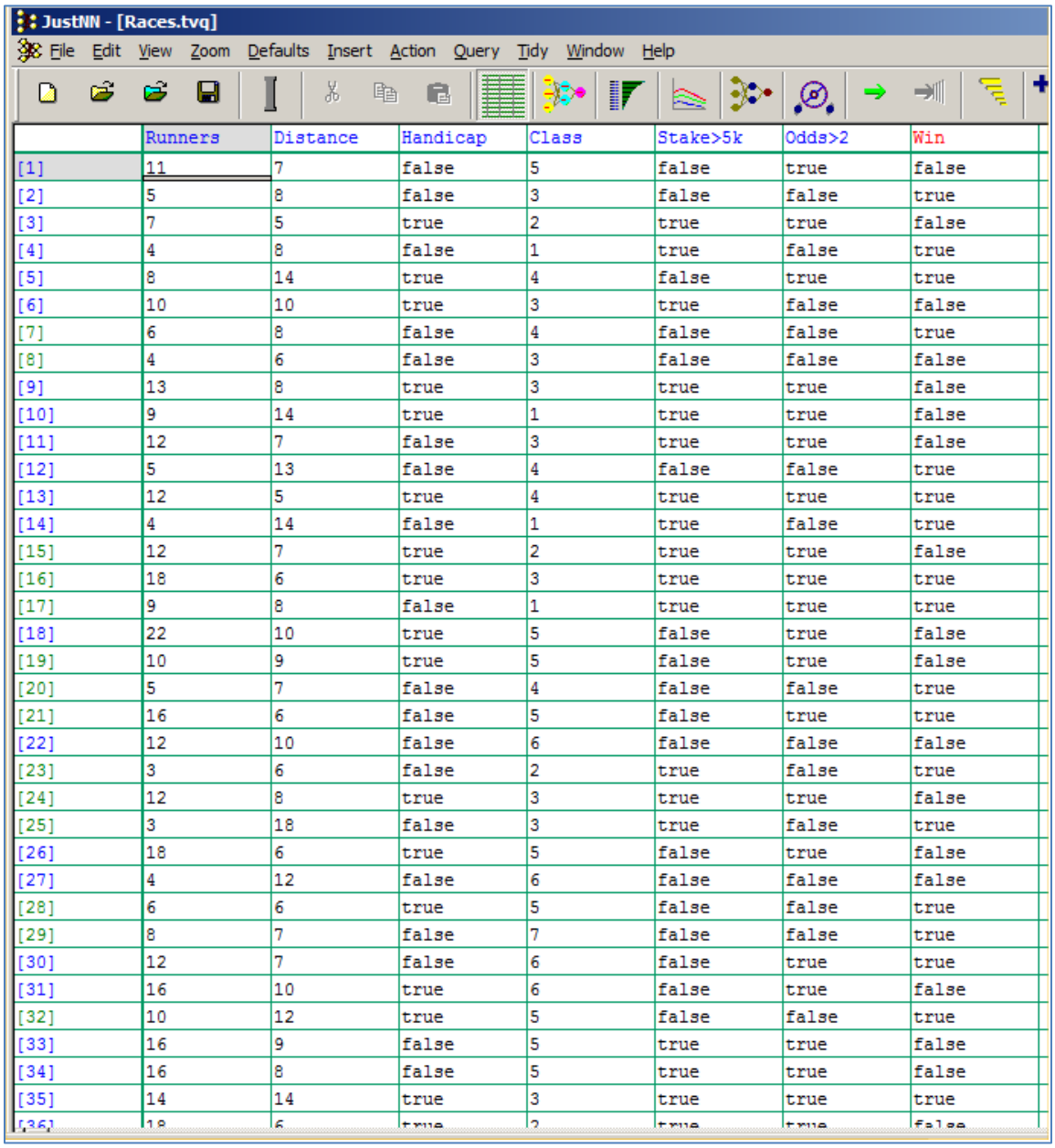
1. XOR. In the first exercise you will open, train and query a simple neural network that simulates XOR, (exclusive-or). XOR is a logical operator that results in the output being true if one of the inputs, but not both, is true. If both inputs are true the output is false.

2. Color Circle. In this exercise you will be guided through a series of steps to make a neural network that learns which secondary color is produced when any two of the three primary colors are mixed together. You will open a partially completed grid file. You will edit the grid and then create, train and query the neural network

3. Races. In the Races exercise you will start with an empty grid. You will then import the results of 370 horse races that **our horse** ran in. Then you will create, train and validate a neural network. When it is completed the neural network will be used to predict the results of other horse races. The columns of data in this example are:

- Runners - how many horses ran in this particular race
- Distance - how long was this race, in furlongs
- Handicap - do better horses carry more weight in order to even out the competition
- Class - higher number means a more competitive or prestigious race (anything from 1 to 6
- Stake>5k - was the entry fee to run in the race greater than $5,000, meaning it is more prestigious
- Odds>2 - were the odds for **our horse** greater than 2-1 odds
- Win - did **our horse** win the race?

Example Data from Races.tvq
(This data can be used for reference when working on Part 4)



	Runners	Distance	Handicap	Class	Stake>5k	Odds>2	Win
[1]	11	7	false	5	false	true	false
[2]	5	8	false	3	false	false	true
[3]	7	5	true	2	true	true	false
[4]	4	8	false	1	true	false	true
[5]	8	14	true	4	false	true	true
[6]	10	10	true	3	true	false	false
[7]	6	8	false	4	false	false	true
[8]	4	6	false	3	false	false	false
[9]	13	8	true	3	true	true	false
[10]	9	14	true	1	true	true	false
[11]	12	7	false	3	true	true	false
[12]	5	13	false	4	false	false	true
[13]	12	5	true	4	true	true	true
[14]	4	14	false	1	true	false	true
[15]	12	7	true	2	true	true	false
[16]	18	6	true	3	true	true	true
[17]	9	8	false	1	true	true	true
[18]	22	10	true	5	false	true	false
[19]	10	9	true	5	false	true	false
[20]	5	7	false	4	false	false	true
[21]	16	6	false	5	false	true	true
[22]	12	10	false	6	false	false	false
[23]	3	6	false	2	true	false	true
[24]	12	8	true	3	true	true	false
[25]	3	18	false	3	true	false	true
[26]	18	6	true	5	false	true	false
[27]	4	12	false	6	false	false	false
[28]	6	6	true	5	false	false	true
[29]	8	7	false	7	false	false	true
[30]	12	7	false	6	false	true	true
[31]	16	10	true	6	false	true	false
[32]	10	12	true	5	false	false	true
[33]	16	9	false	5	true	true	false
[34]	16	8	false	5	true	true	false
[35]	14	14	true	3	true	true	true
[36]	18	6	true	2	true	true	false

Part 4 - Majority Learning in a Neural Net

In this part of the lab project, do the following, and include results in your lab report.

1. Create a new JustNN document. Open the RacesEmpty.tvq file after downloading it from the class Schedule page.

2. Starting with the one empty training row, define 10 instances of 7-column training examples, with values similar to those in the Races exercise data. To create a new instance, click Insert->Training Example Row. Fill in any values you like for each of the columns (or "attributes"). Repeat that 10 times to create the 10 new rows.

3. Select 2 out of the 10 examples you just added for validation. For each of those two rows, double-click on the start of the row, hit Enter, then in the dialog that pops up change the row type from "Training" to "Validating", then click OK.

4. Save it.

5. Click the "Grow New Network" button, which will lead you through several steps of the **backpropagation** learning process. On the "Controls" dialog, set the "Fixed period steps" to "Stop on" 1000 cycles.

6. Include a screen shot of your trained network in the lab report. Did the network make any inputs weigh much more than others? In your report, describe whether you see any relationship between the stronger weights and the 1s and 0s (true and false values) of the examples you trained with.

7. Include a screen shot of the learning curve for the network in the lab report. In your report, explain why your network's curve looks the way it does. It will probably have poor performance. Look at your examples and report whether the choice of examples you created and the specific validation examples had an impact on the results and why?

8. Create three more of the 7-column example rows that have different values from the earlier rows. In two of the rows, leave 2 of the input values ("attributes") empty, and for the third leave 4 of the input values empty. Use them as Query examples in your grid. See what answers the network gives for these examples. List these examples in your write-up, and tell what answers the network gave. Do the answers make sense, given the values you did fill in? If not, why not? Are these examples in some way too different from the training examples?

9. Make another 14 examples (don't use the three you made in #8, set them as "Validating"), but make sure there are seven where the output should be false and seven where the output should be true. Put these in your report. Do you get a better learning curve when you have 24 training examples? How do your 3 queries work out now? Much better, or only marginally better than before?

Part 5 - Image Recognition

In this part of the lab, you will design a neural network to recognize images of $1, $5, and $10 (and possibly other) bills. This technology you will develop could be the basis for a bill reader in a vending machine.

Note: There is no UNDO feature in JustNN, so be careful!

For your report, including copious screen grabs to document what you do and your responses to the questions (indicated with a "Q") below.

1. Save and extract the **lab5bills.zip** file from the class Schedule page, which creates a folder called **lab5bills**. Save the file and the folder you extract someplace convenient for easy access.

Experiment 1

2. From JustNN, open the **BillDetector.tvq** file. It contains references to the images in the image files in the **lab5bills** folder. The images are all 128 pixels (dots) across, and 54 pixels down. Double-click on one of the image names (e.g., "one - 1"), check the "Show all images in grid" box, then click OK.

   Troubleshooting: If the images do not appear in the grid, you need to correct this. Double-click on the first image name for which no image appears, then click the "Image File" button. Note the file name in the box, then click the "Browse" button, locate the file of the same name, and "Open" it. The reset of the images should now appear in the grid.

3. Look at the Grid. It will have 18 rows (12 examples, 6 validating). Each example and validating row has a single input (an image), and three Boolean attributes (first column = $1, second column = $5, third = $10) that are set true or false depending on which type of bill is on that row. Note there are also 7 query rows at the bottom of the grid, with fall values for all three attributes.

4. Explore the grid ( ) and network ( ) views to see how things are starting out.

   Q1. Describe the initial "state" of the grid and neural network, and include a screen grab of the network.

5. Make a network from this data ( ). You'll see that even though there is one input for examples, the network will have THREE input nodes. This is because for pictures, JustNN looks at all the pixels (color dots) in the picture and calculates for us 3 values: PC (pair code), EC (edge code), and BC (block code).

   - PC is based on the sum of differences of the first 1000 pairs of pixels in the image.

- EC is based on the locations of the sharpest edges on the top, bottom, left, and right sides of the image, and
- BC is based on the location of the largest block of same-color pixels in the image.

6. Follow the prompts to train the net. First, in the "New Network" dialog make sure all three hidden layers are unchecked (i.e., Grow layer number 1, 2 and 3 are unchecked), and click "OK." Next, click "Yes" to set the controls.

   In the "Controls" dialog, check the box in the lower right area that says "Stop on" and have it stop on "1000" cycles. Click "OK", "Yes" and "OK" to have the network start learning.

7. Once learning is complete, which will happen quickly, observe the query rows. In each query row, double-click on each "true" and "false" to see how confident the network is about each answer (on a scale from 0.0 to 1.0). False is 0.0, True is 1.0, and anything else in between falls somewhere on the range from false to true.

   Q2. Which input features in this neural network (PC, EC and BC) had stronger weights? Include a screen grab of the network and of the learning curve ( ).

   Q3. How was the learning performance? What did the network do well at learning to recognize? What sort of problems with learning did you observe, such as bills that were misrecognized as an incorrect value, etc.? Briefly explain why you think this happened.

8. Once complete with Experiment 1, close the network (without saving changes) and exit JustNN. Do this at the conclusion of each experiment to ensure that each one is a "clean" experiment and the results are not affected by the previous experiment.

Experiment 2

9. Two possible reasons the network might have learning problems in the first experiment are that there were no hidden layers in the network and training was done for a very short time.

10. Repeat the first experiment three more times as follows:

    - First, open the **BillDetector.tvq** file in JustNN.
    - Grow the network as before ( ) only this time include one hidden layer consisting of a maximum of 7 nodes and learning for 1000 cycles. Observe the results.

Q4. How did learning differ with this slightly different approach that used a hidden layer? Why do you think learning was different? Include screen grabs of anything that helps to explain this difference, such as the network or learning curve.

- Second, grow the network again ( ) with one hidden layer (max 7 nodes) and have it "Stop after" after 20.0 seconds. Be sure to uncheck the "Stop on" checkbox. Observe the results.

Q5. How did learning differ with this approach that used a hidden layer with a much longer training period? Why do you think learning was different? Include screen grabs of anything that helps to explain this difference, such as the network or learning curve.

- Third, grow the network a third time ( ) with all three hidden layers (7, 4, and 4) and have it stop after 60.0 seconds. Observe the results.

Q6. How did learning differ with this approach using three hidden layers and an even longer training period? Why do you think learning was different? Include screen grabs of anything that helps to explain this difference, such as the network or learning curve.

Experiment 3

11. Try growing a new network as you did for Experiment 2 one more time. This time, you may select any number of hidden layers, the maximum sizes of the layers, and any other learning parameters you would like to experiment with. Explore in any way you like, making sure to note what approach you used. Before starting, make a note of what you think the result will be.

Q7. Describe your hypothesis for the approach you are going to use. What do you think will happen?

Q8. Explain the approach you used, how well it learned compared to other approaches you tried earlier, and why you think it differed. As usual, include any screen grabs that help to illustrate your results.

Experiment 4

12. Determine the effect, if any, of different amounts of training. Select one of the above network configurations and repeat learning "Stop after" 30, 60 and 120 second durations.

Q9. What differences did you see with learning performed for different durations? Are there benefits or tradeoffs in longer periods of training?

Experiment 5

13. This sort of neural network could be used in a commercial vending machine, bill change machine or bill counterfeit detection machine.

    Q10. How would you incorporate such a neural network into one of these machines? Describe in general (non-technical) terms how the neural network could be used to identify denominations of bills, when it would reject bills and how accurate it could be.

    Q11. Suppose we wanted our network to reject non-$1,$5, and $10 bills from being processed (maybe our vending machine can't store enough change for these!). What change would you make to the network (e.g. entries in the Grid) to accomplish this?

Experiment 6

14. Test your preferred neural network bill detector on three additional bills. To do this, take photos using a phone-camera or download images from the Internet. You will likely need to crop and resize the images you gather so they are each 128x54 pixels in size. There are numerous online tools that can help with this, such as Pic Resize or others found via a Google search for "online image crop and resize".

    Add these new pictures into your neural network by first copying them into the same directory (lab5bills), and then adding them one by one as new query rows. To add additional query rows, do the following:

    - Select Insert->Querying Example Row, then click OK.
    - Double-click in the Image Name cell and type a name for the image (e.g., "query 1"), and click OK.
    - Click the Browse button and select the image you want to associate with this query row, and click Open.

    Once the new query rows are added, view the results for each query, which happens automatically if the network has already been trained. Observe the results.

    Q12. How accurate was the detector at identifying these additional bills? If there were misidentifications, explain what you think the reasons are. Include the new images you used in your report.